

## 发布订阅模式

发布订阅模式, 又叫观察者模式, 它定义对象间一种一对多的依赖关系, 当一个对象的状态发生改变时, 所有依赖与它的对象都将得到通知

构成: 发布者, 订阅者, 处理者

### 实现一个发布订阅模式

```
class Event {
  constructor () {}
  // 首先定义一个事件容器, 用来装事件数组 (因为订阅者可以是多个)
  handlers = {}

  // 事件添加方法, 参数有事件名和事件方法
  addEventListener (type, handler) {
    // 首先判断handlers内有没有type事件容器, 没有则创建一个新数组容器
    if (!(type in this.handlers)) {
      this.handlers[type] = []
    }
    // 将事件存入
    this.handlers[type].push(handler)
  }

  // 触发事件两个参数 (事件名, 参数)
  dispatchEvent (type, ...params) {
    // 若没有注册该事件则抛出错误
    if (!(type in this.handlers)) {
      return new Error('未注册该事件')
    }
    // 遍历触发
    this.handlers[type].forEach(handler => {
      handler(...params)
    })
  }

  // 事件移除参数 (事件名, 删除的事件, 若无第二个参数则删除该事件的订阅和发布)
  removeEventListener (type, handler) {
    // 无效事件抛出
    if (!(type in this.handlers)) {
      return new Error('无效事件')
    }
    if (!handler) {
      // 直接移除事件
      delete this.handlers[type]
    } else {
      const idx = this.handlers[type].findIndex(ele => ele === handler)
      // 抛出异常事件
      if (idx === undefined) {
        return new Error('无该绑定事件')
      }
      // 移除事件
      this.handlers[type].splice(idx, 1)
    }
  }
}
```

```
        if (this.handlers[type].length === 0) {  
            delete this.handlers[type]  
        }  
    }  
}
```